

*Final Report***PHYSICS-BASED PROGNOSTIC TOOLS FOR FATIGUE AND CREEP-FATIGUE OF
PROPULSION SYSTEM MATERIALS**David L. McDowell^{1,2}Yustianto Tjiptowidjojo, Graduate Student¹Craig Przybyla, Graduate Student²¹G.W. Woodruff School of Mechanical Engineering²School of Materials Science & Engineering

Georgia Institute of Technology, Atlanta, GA 30332-0405

Yustianto Tjiptowidjojo, Graduate Student

Craig Przybyla, Graduate Student

Technical Monitor: "Grelotti, Robert A." robert.grelotti@pw.utc.com, Pratt & Whitney

December 2007

Table of Contents

Executive Summary	1
1. Material Point Simulator (MPS) and the User Material (UMAT)	2
2. Artificial Neural Network (ANN)	4
ANN Training of the Macroscale ISV Model Parameters	4
3. UMAT and Finite Element Analysis	8
Creating input file for the ANSYS UMAT	9
Summary of Steps Needed to Compile and Link ANSYS User Material Subroutines (UserMat)	12
4. Pratt and Whitney Microstructures	13
Task 1. Run a single node to 2% Strain at 1200 °F (650°C)	16
Task 2. Strain vs. Time plots (2% - 5% max) for	17
Task 3. Stress vs. Time plots for 5% strain for a simple bar	17
Task 4. Simulations using Pratt & Whitney component	19
Task 5. Fatigue Indicator Parameters	23
5. Development of Ti-6Al-4V Macroscale Model	25

Executive Summary

This report details progress to date of this project from its inception in January 2007 to closing date of December, 2007. Section 4 provides the results of the requested simulations for the Pratt & Whitney component. The objectives on implementation of microstructure-sensitive model for IN 100 are fully met. Section 5 on Ti-6Al-4V describes progress in our efforts to develop a microstructure-sensitive macroscopic ISV model for this alloy, including texture effects.

1. Material Point Simulator (MPS) and the User Material (UMAT)

The subject at hand in this work is the implementation of the macroscale internal state variable (ISV) model with microstructure-sensitive parameters based on microstructure-sensitive crystal plasticity models (see constitutive model in **Appendix A** for single crystal model of Ni-base superalloys). This involves using a Material Point Simulator (MPS) to integrate uniaxial relations from experiments and polycrystal plasticity simulations. The MPS evaluates the response of material points subjected to homogeneous imposed deformation histories, and employs the 1-D formulation of the ISV model, with the formulation given in **Appendix B**. The implementation of the MPS is given in **Figure 1**.

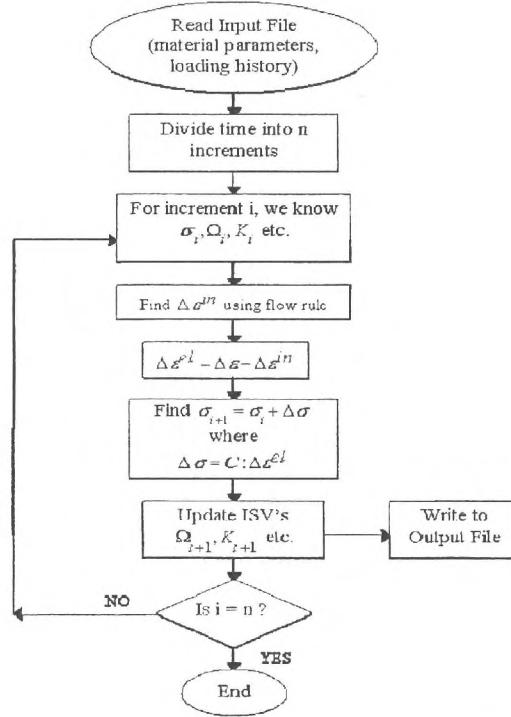


Figure 1. Flowchart of the MPS for uniaxial macroscale ISV relations.

For polycrystal plasticity of Ni base superalloys, it is assumed that material point consists of a collection of crystals (grains) whose individual mechanical responses are linked to that of the material point using the extended Taylor hypothesis; a User Material Subroutine (UMAT) is called for each grain. This crystal plasticity UMAT implements an integration procedure for all possible slip systems in 3-D. For α - β Ti alloys, it is necessary to conduct explicit finite element modeling of a set of grains comprising the polycrystal owing to the low symmetry of the HCP α phase.

Ultimately, a UMAT that implements the 3-D formulation of the ISV model (Walker-type) is suitable for structural analysis using finite element codes such as ABAQUS and ANSYS. The implementation details are given in **Appendix C**. The UMAT can then be used in Finite Element Analysis (FEA) to compute the mechanical response of Ni-base superalloy (nominally IN100) structures with spatial variation of heat treatment and therefore precipitate distributions and grain sizes. It is first necessary to find the corresponding material parameters based on the underlying microstructures. A scheme has been developed to correlate the UMAT macroscale ISV model material parameters with the microstructure attributes; the relevant flowchart is given in **Figure 2**. An artificial neural network (ANN) has been trained with a sufficient number of experiments and polycrystal plasticity simulations over a range of microstructures to relate the ISV model parameters in the to the corresponding microstructure parameters. Once the ANN is trained, it can be used to predict the material parameters for the macroscopic ISV model for any given microstructure. The ANN is not computationally intensive. The parameter determination employs ISIGHT 9.0 optimization software. In short, the MPS material parameters can be obtained from any given microstructure within the range of those used for training (experiments and simulations) via ANN. Next, the MPS material parameters can be converted to the UMAT material parameters as outlined in Section 3.

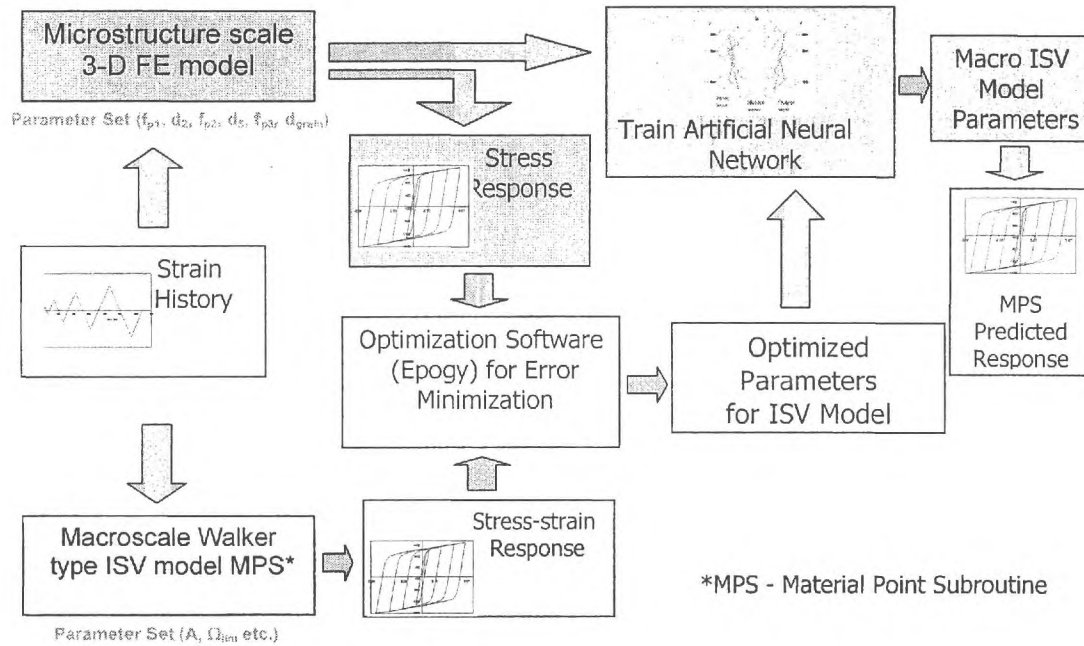


Figure 2. Flowchart for linking simulations from the polycrystal plasticity model and experiments for various IN100 microstructures to the macroscale ISV model (or 'Walker' model).

2. Artificial Neural Network (ANN)

ANN Training of the Macroscale ISV Model Parameters

The Matlab 7.0 Neural Network Toolbox was used to train the Artificial Neural Network with two sets of data. The first set contains the microstructure attributes, such as: volume fraction of primary γ' precipitates, f_{p1} ; volume fraction of secondary γ' precipitates, f_{p2} ; secondary γ' precipitate size, d_2 ; volume fraction of tertiary γ' precipitates, f_{p3} ; tertiary γ' precipitate size, d_3 ; and mean grain size, d_{grain} . The second set contains the corresponding ISV model parameters, such as: ratio of the hardening rate and the saturation limit of the first back stress component, $A^1/\Omega_{\text{lim}}^1$; the saturation limit of the first back stress component, Ω_{lim}^1 ; hardening rate of the second back stress component, A^2 ; Elastic Modulus, E ; threshold stress, κ ; the saturation limit of the isotropic hardening, Q_h ; and the contribution from the rate dependent term, A_{v2} .

Important commands in the Matlab 7.0 Neural Network Toolbox

PRESTD	:	Normalize data for zero mean and unity standard deviation
PREPCA	:	Principal components analysis
NEWFF	:	Initializes feed-forward networks
TRAIN	:	Trains a network
SIM	:	Simulates networks
POSTSTD	:	Inverts PRESTD to convert network outputs to original units
POSTREG	:	Linear regression between targets and trained network outputs

Using the commands above, a feed-forward network is trained to perform a nonlinear regression between the microstructural properties and the corresponding macroscale ISV model parameters. The final network is analyzed to investigate the overall performance.

In this example, we are going to find the correlation between the microstructure arrangements and their effects on the κ ; the saturation limit of the isotropic hardening, Q_h . This is done through a file called “trial_Q.m”.

I. DEFINING THE PROBLEM

The array “input_data.xls” contains the microstructural arrangements of the 150 microstructures that have been previously fitted to obtain the 150 corresponding sets of the Macroscale ISV model parameters. The array “target_data.xls” contains the 150 sets of the Macroscale ISV model parameters. We first need to load in the data file, which is done through the following commands:

```
A=xlsread('input_data.xls');  
p=transpose(A);
```



```

B=xlsread('target_data.xls');
Bt=transpose(B);
t=Bt(6,:)

```

Here, Matlab is dealing with two sets of data, p and t . The first set, p , contains the microstructural arrangements of the 150 microstructures. The second set, t , contains the corresponding Q_h for those microstructures, which is located at the 6th entry in “target_data.xls”.

Next, we normalize the inputs and targets so that they have zero mean and unity variance.

```
[pn,meanp,stdp,tn,meant,stdt] = prestd(p,t);
```

The command ‘prestd’ preprocesses the network training set by normalizing the inputs and targets so that they have means of zero and standard deviations of 1. prestd(p,t) takes these inputs,

p : $R \times Q$ matrix of input (column) vectors
 t : $S \times Q$ matrix of target vectors

and returns

pn : $R \times Q$ matrix of normalized input vectors
 $meanp$: $R \times 1$ vector containing mean for each P
 $stdp$: $R \times 1$ vector containing standard deviations for each P
 tn : $S \times Q$ matrix of normalized target vectors
 $meant$: $S \times 1$ vector containing mean for each T
 $stdt$: $S \times 1$ vector containing standard deviations for each T

Next, perform a principal component analysis and remove those components which account for less than 0.1% of the variation.

```
[pttrans,transMat] = prepca(pn,0.001);
```

The command ‘prepca’ preprocesses the network input training set by applying a principal component analysis. This analysis transforms the input data so that the elements of the input vector set will be uncorrelated. In addition, the size of the input vectors may be reduced by retaining only those components which contribute more than a specified fraction (min_frac) of the total variation in the data set. prepca(P,min_frac) takes these inputs

P : $R \times Q$ matrix of centered input (column) vectors
min_frac : Minimum fraction variance component to keep

and returns

ptrans : Transformed data set
transMat : Transformation matrix

Next, we divide the data up into training, validation and test sets. The testing set will start with the second point and take every fourth point. The validation set will start with the fourth point and take every fourth point. The training set will take the remaining points.

```
[R,Q] = size(ptrans); iitst = 2:4:Q;  
iival = 4:4:Q; iitr = [1:4:Q 3:4:Q];  
vv.P = ptrans(:,iival); vv.T = tn(:,iival);  
vt.P = ptrans(:,iitst); vt.T = tn(:,iitst);  
ptr = ptrans(:,iitr); ttr = tn(:,iitr);
```

II. DEFINING THE NETWORK

We are now ready to define the network. We create a feedforward network with 5 hidden neurons, 3 output neurons, TANSIG hidden neurons and linear output neurons. Here we assign the Levenberg-Marquardt training function - TRAINLM. You can replace TRAINLM with any training function you desire; this can be found in the Matlab 7.0 ANN Toolbox. The NEWFF command will also initialize the weights in the network.

```
net = newff(minmax(ptr),[3 1],{'tansig' 'purelin'},'trainlm');
```

In Matlab 7.0, net = newff creates a new network with a dialog box; newff(PR,[S1 S2...SNI],{TF1 TF2...TFNI},BTF,BLF,PF) takes

PR : R x 2 matrix of min and max values for R input elements
Si : Size of ith layer, for NI layers
TFi : Transfer function of ith layer, default = 'tansig'
BTF : Backpropagation network training function, default = 'traingdx'
BLF : Backpropagation weight/bias learning function, default = 'learnngdm'
PF : Performance function, default = 'mse'

and returns an N layer feed-forward backpropagation network. The transfer functions TFi can be any differentiable transfer function such as tansig, logsig, or purelin. The training function BTF can be any of the back propagation training functions such as trainlm, trainbfg, trainrp, traingd, etc. Caution: trainlm is the training function we use because it is very fast, but it requires a lot of memory to run.

III. TRAINING THE NETWORK

If we do not wish to see the results while training, we can issue the command

```
net.trainParam.show = NaN;
```

Next, we train the network. We use early stopping, so we are passing the validation data. We also want the errors computed on a test set, so we are passing the testing data. The validation and testing data are named vv and vt, respectively.

```
[net,tr]=train(net,ptr,ttt,[],[],vv,vt);
```

IV. TESTING THE NETWORK

We can plot the training, validation and test errors via the commands:

```
plot(tr.epoch,tr.perf,'r',tr.epoch,tr.vperf,'g',tr.epoch,tr.tperf,'-b');
legend('Training','Validation','Test',-1);
ylabel('Squared Error')
```

Once the network has been trained, it can be simulated using the following command:

```
an = sim(net,ptrans);
```

Next, it is now necessary to convert the output of the network back into the original units of the targets. Since the targets were transformed using PRESTD so that the mean was 0 and the standard deviation was 1, we need to use POSTSTD (the inverse of PRESTD) and the original mean and standard deviation to transform the network outputs back into the original units.

```
a = poststd(an,meant,stdt);
```

V. DISPLAY RESULTS

We will now display plots showing regression analyses between the network outputs and the corresponding targets (in original units) by using the input:

```
for i=1:l
    figure(i+1)
    [m(i),b(i),r(i)] = postreg(a(i,:),t(i,:));
end
```

In the first plot, it is desired that the training, validation, and test to lie on the same curve, which implies that ANN successfully correlates the input and output data. In the second plot, it is desired that the predicted values (A) match the target values (T) so as to form a perfect linear fit. The procedure of training ANN can be repeated several times until the desired condition is met and then saved into a Matlab data file. This method can be generalized to correlate the microstructural arrangements to other macroscale ISV model parameters.

1. Interpolating values using ANN

In this example, the saved Matlab data file is named “matlab_Q.mat”. To interpolate the ISV model parameters using the trained ANN, we must first define the microstructures that we are interested in. In this example, it is in the file called “Micros_of_Interest.xls”. It contains all the desired microstructural arrangements whose Macroscale ISV model parameters we need to find.

Next, we double click the Matlab data file “matlab_Q.mat” and notice that the trained ANN is loaded into Matlab. To find the ISV model parameters for the desired microstructures, we type

```
p2p=xlsread('Micros_of_Interest.xls')
p2=transpose(p2p)
[p2n] = trstd(p2,meanp,stdp)
[p2trans] = trapca(p2n,transMat)
an = sim(net,p2trans);
[a2] = poststd(an,meant,stdt)
a2t=transpose(a2)
xlswrite('Micros_of_Interest_Predicted_Q.xls', a2t)
```

The command TRASTD preprocesses the network training set using the mean and standard deviation that were previously computed by PRESTD. This function needs to be used when a network has been trained using data normalized by PRESTD. All subsequent inputs to the network need to be transformed using the same normalization. The command TRAPCA preprocesses the network input training set by applying the principal component transformation that was previously computed by PREPCA. This function needs to be used when a network has been trained using data normalized by PREPCA. All subsequent inputs to the network need to be transformed using the same normalization.

The corresponding macroscale ISV model parameters are then written into a file called “Micros_of_Interest_Predicted_Q.xls”. This method can be repeated for the other macroscale ISV model parameters.

3. UMAT and Finite Element Analysis

Parameter Conversion

The macroscale ISV model parameters obtained from the ANN can now be used to predict the stress-strain behavior of a complex geometry with various IN100 microstructures. However, before we do that, we need to convert those material parameters into the form that are required by the UMAT. The Macroscale ISV model parameters from the ANN are still in the 1D (MPS) format since that was the formulation that was used to link the microstructures to the macroscale ISV model parameters, based on uniaxial cyclic loading experiments. The conversion can easily be performed

according to the conventions defined in Table 1. Other parameters with the same names are identical in both formulations.

Table 1. Conversion from 1-D to 3-D Parameters

<i>3-D Parameters</i>	<i>1-D Parameters</i>
A_1	A^1
A_2	A^2
B_1	$A^1/\Omega_{\text{lim}}^1$
B_2	$A^2/\Omega_{\text{lim}}^2$
E_1	$b_h Q_h$
F_1	b_h

Creating input file for the ANSYS UMAT

Below is an example of a sample input file which calls the user material subroutine linked to the current compilation of ANSYS. For instructions on linking a user material subroutine to ANSYS see *Linking the UMAT to ANSYS* below. This file essentially contains various scripting commands inherent to ANSYS to set up the model, define the material properties, run the simulation, and then post process the data.

```
!!*****
!! This is an input file that contains a block under uniaxial tension
!! for ANSYS UMAT test
!!*****

Finish          ! Exit all processors

/Title, Walker_block

!!*****

/Prep7          ! To begin preprocessing (define the model)

!! Create the geometry
block,0,1,0,1,0,1
!! This creates a block with the first node at (0,0,0) at the second at
!! (1,1,1)

!! Name elements and materials
ele1=185
mat1=1

!! Define the element type
et,1,ele1      ! Linear 8-Node Brick Element

! Mesh the geometry
mshkey,1      ! Force a mapped mesh
lesize,all,,,1
```

```

!! Specify size control, put one element along each edge
vmesh,all          ! Generate the mesh

!!*****
!! Define the material properties
!! This input contains the material properties for linear isotropic
!! materials and the parameters defined by the Walker Viscoplastic
!! Model coded in the UserMat. Comment on/off the one that you want to
!! use

!! For linear isotropic material
!mp,ex,1,22798174.31
!! Material property, Young's Modulus = 22798174.31 psi

!mp,nuxy,1,0.33
!! Material property, Poisson's Ratio = 0.33

!! For the material parameters to be used with the UserMat
!! User Material Type = 3D/Axisymmetric Walker Viscoplastic Model

tb,user,mat1,1,17
!! Activate a data table for nonlinear material properties or special
!! element input. This command states that this input file uses a user-
!! defined material model, with mat1 as the set of material properties,
!! with 1 number of temperature provided, and 17 material constants

tbtemp,1.0
!! Define a temperature for the data table. It is set equal to 1, but
!! really has no influence in the calculations

!! Material Parameters for the Walker Viscoplastic Model are inserted
!! below. Constants=17 (This is for Flange) (UNITS ARE IN ENGLISH)

!! Input the material parameters 1-6
!! Av, bv, expN, a_kappa, A1, B1
tbdata,1,0.00125,0.0,95.,82372.63297,31648045.69,1179.29

!! Input the material parameters 7-12
!! E1, F1, R0, D, Emod, Enu
tbdata,7,68169.4189,6.5,0.0,36258.15809,22798174.31,0.33

!! Input the material parameters 13-17
!! A2, B2, Av2, Bv2, expN2
tbdata,13,555545.0272,9.576,115611.52,0.0,7.0

tb,state,mat1,,31
!! Define material = mat1, that has 31 state variables
!!*****

Finish          ! Exit all processors

!!*****
/solu          ! Enter the solution preprocessor

!! Define the boundary conditions
asel,s,loc,y,0    ! Select area(s) on bottom of cube (i.e. at y = 0)
da,all,uy,0

```



```

!! Constraint the displacement in the y direction (uy=0)
allsel                ! Reselect everything, back to the whole area

asel,s,loc,x,0        ! Select area(s) on left face of cube (i.e. at x = 0)
da,all,ux,0
!! Constraint the displacement in the x direction (ux=0)
allsel                ! Reselect everything, back to the whole area

asel,s,loc,z,0        ! Select area(s) on back of cube (i.e. at z = 0)
da,all,uz,0
!! Constraint the displacement in the x direction (uz=0)
allsel                ! Reselect everything, back to the whole area

!! Define the loading condition

nlgeom,on
!! Include large-deflection effects in a static or full transient
!! analysis.

asel,s,loc,y,1        ! Select the top face of the cube (y=1)
antype,0              ! Specify the analysis type and restart status
deltim,1e-01,1e-03,2e-01
!! Specify the time step sizes to be used for this load step
!! Time step size for this step = 5e-02, Minimum time step = 1e-03
!! Maximum time step = 2e-01

da,all,uy,0.1
!! Set uy=0.1 on this selected area (i.e. eng_strain_yy = 0.1)
time,10.0
!! Sets the time for a load step. In this case, t=10.0 sec.

allsel                ! Reselect everything, back to the whole area

outres,,all
! Control the solution data written to the database.
outpr,all,all         ! Controls the solution printout

! Solve the problem
/solu
solve                 ! Solve it
!!*****

Finish                ! Exit all processors

!!*****
!! Post-Processing
/post26               ! Enter the time-history results postprocessor

!! Specify element data to be stored from the results file
esol,2,1,,s,x,SX
esol,3,1,,s,y,SY
esol,4,1,,s,z,SZ
esol,5,1,,epel,x,EX
esol,6,1,,epel,y,EY
esol,7,1,,epel,z,EZ
esol,8,1,,eppl,x,EX
esol,9,1,,eppl,y,EY

```

```

esol,10,1,,eppl,z,EZ

! List variables vs. time (or frequency).
prvar,2,3,4
prvar,5,6,7
prvar,8,9,10

!!*****

! Finish                      ! Exit all processors

!!*****

```

1. Linking the UMAT to ANSYS

Summary of Steps Needed to Compile and Link ANSYS User Material Subroutines (UserMat)

Note: This has been implemented for ANSYS Release 10.0 on a system running Windows XP. Please reference the ANSYS supplied documentation entitle: "Guide to ANSYS User Programmable Features, ANSYS Release 10.0, Aug. 2006.

1. Locate directory in **C:\Program Files\Ansys Inc\v100\ansys\custom\user\<platform>** (here <platform> is a directory that uniquely identifies the hardware platform version.)
2. Save a copy of this directory in the same user folder (**C:\Program Files\Ansys Inc\v100\ansys\custom\user**) under any desired name. **DO NOT MODIFY FILE IN ORIGINAL DIRECTORY!** This is your backup.
3. In the copied version of the <platform> folder delete any unnecessary C++ or Fortran supplied programmable features not necessary to the current desired compilation of Ansys. (i.e. in this case we just want usermat.f containing the desired material model and we can delete any other files ending in ".f" or ".c")
4. Save the desired version of usermat.f in the copied version of the <platform> folder.
5. The required Fortran compiler for ANSYS Release 10.0 on Windows based systems is the current Visual Fortran Compiler by Intel Release 8.0 coupled with Microsoft Visual Studio 2003 (MVS2003). It is highly recommended to use the Command Prompt window set up by the installation of Visual Fortran to compile the ANSYS code with user subroutines, otherwise an error will occur when the process initiates the linker. To open this prompt initiate the following sequence under the Windows XP start menu: **Start→All programs→Intel(R) Software Development Tools→Intel(R) Fortran Compiler 8.x→Build Environment for Fortran IA-32 applications.**
6. In the command prompt make sure the path is set to the copied version of the <platform> folder with the correct usermat.f file.
7. Run anscust.bat.
8. When asked: "Do you want to link with the small export list?" it is advised to answer no.

9. The compiler will compile all the user supplied routines located in the directory ending in either “.f”, “.for”, “.c” or “.cpp” and linker will then create a custom executable called ANSYS.exe in your working directory. It is possible to rename the executable after creation.
10. Bring up the ANSYS Interactive Launcher.
11. Click the button with the ellipsis (“...”) next to the option “Execute a customized ANSYS executable”. Select the .exe you have just created.
12. Start ANSYS.

If your executable is running properly, ANSYS will print “This ANSYS version was linked by Licensee” in the output window.

2. FEA Simulation in ANSYS using the UMAT

To run the input file with calls the UMAT start the customized ANSYS executable using the ANSYS Interactive Launcher as described above in *Linking the UMAT to ANSYS*. On the main menu bar in ANSYS select **File**→**Read input from...** and select the location of the text file containing the appropriate material definitions as defined by *tbdata* (see *Creating of ANSYS input file for the UMAT* above).

4. Pratt and Whitney Microstructures

The microstructure information supplied by Pratt and Whitney is given in **Table 2** for several components.

Table 2. PWA 1100 Microstructure Information.

Sample	Primary γ'		Secondary γ'		Tertiary γ'		Grain Size (μm)
	Ave. Size (μm)	V_f	Ave. Size (nm)	V_f	Ave. Size (nm)	V_f	
“Rim”	1.18	0.20	141.0	N/A	7.55	N/A	4.1
“Bore Center”	1.24	0.21	195.0	0.336	8.61	N/A	4.1
“Flange”	1.16	0.219	145.0	0.33	9.98	N/A	3.6

The Artificial Neural Network (ANN) was trained and calibrated within the ranges of microstructure features given in **Table 3**.

Table 3. Ranges of microstructure features used to generate the database for 150 microstructures.

Microstructure Attribute	Min	Max
Volume Fraction of Primary γ' Precipitate, f_{p1}	0.0	0.25
Volume Fraction of Secondary γ' Precipitate, f_{p2}	0.29	0.46
Secondary γ' Precipitate size, d_2 (nm)	110	340
Volume Fraction of Tertiary γ' Precipitate, f_{p3}	0.024	0.14
Tertiary γ' Precipitate size, d_3 (nm)	7.0	21.0
Grain Size, d_{grain} (microns)	4.0	34.0

* The primary γ' precipitate size, d_1 , was set to 1 μm .

The supplied microstructure “Rim” does not fall within the ranges of microstructure features used to train the ANN since it does not contain any secondary γ' precipitate or its volume fraction is not known so at this point it has been ignored. The macroscopic model parameters from the trained ANN for “Bore Center” and “Flange” were obtained assuming that the volume fraction of the tertiary γ' precipitates fell within the ranges given in **Table 3**. Making this assumption, the given microstructure features were then modified according to **Table 4**.

Table 4. Modified PWA 1100 Microstructure features.

Sample	Primary γ'		Secondary γ'		Tertiary γ'		Grain Size (μm)
	Ave. Size (μm)	V_f	Ave. Size (nm)	V_f	Ave. Size (nm)	V_f	
“Bore Center”	1.0*	0.21	195.0	0.336	8.61	0.024*	4.1
“Flange”	1.0*	0.219	145.0	0.33	9.98	0.024*	3.6**

The numbers labeled with * are the modified values. Note that the “Flange” grain size also does not fall within the ANN range; however, the value is within the vicinity of the lower bound.

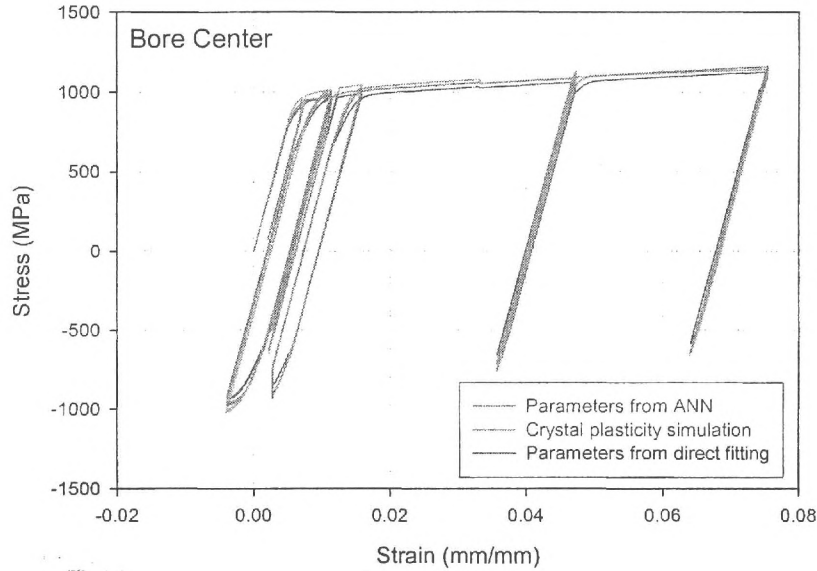
The macroscopic model parameters for “Bore Center” and “Flange” obtained using the trained ANN and are tabulated in **Table 5**, based on experiments and simulations at a temperature of 650°C. A set of parameters were assigned for the model for all microstructures, namely $N = 95$, $N_2 = 7$, $D = 250$ MPa, $A_v = 1.25 \times 10^{-3} \text{ s}^{-1}$ and $F_I = 6.5$.

Table 5. Macroscopic viscoplastic model parameters obtained from ANN (650°C).

Sample	κ (MPa)	A_1 (GPa)	B_1	E_1 (MPa)
“Bore Center”	525	206	1182	446
“Flange”	568	218	1179	470

Microstructure	E_{mod} (GPa)	ν	A_2 (MPa)	B_2	$A_{v_2} \times 10^{-15}$ (s ⁻¹)
“Bore Center”	157.2	0.33	2887	7.2	147191
“Flange”	157.2	0.33	3830	9.6	115612

To check the validity of these parameters, **Figures 3 and 4** show the comparison of the stress-strain curves under a random uniaxial strain history for a material point using the macroscale ISV model with parameters are obtained from ANN, and two other material point simulations with parameters obtained directly from fitting the macroscale ISV model to a polycrystal plasticity simulation for two microstructures (Bore Center and Flange) and the polycrystal plasticity model itself. The agreement is noteworthy.

**Figure 3.** Stress-strain comparison for Bore Center.

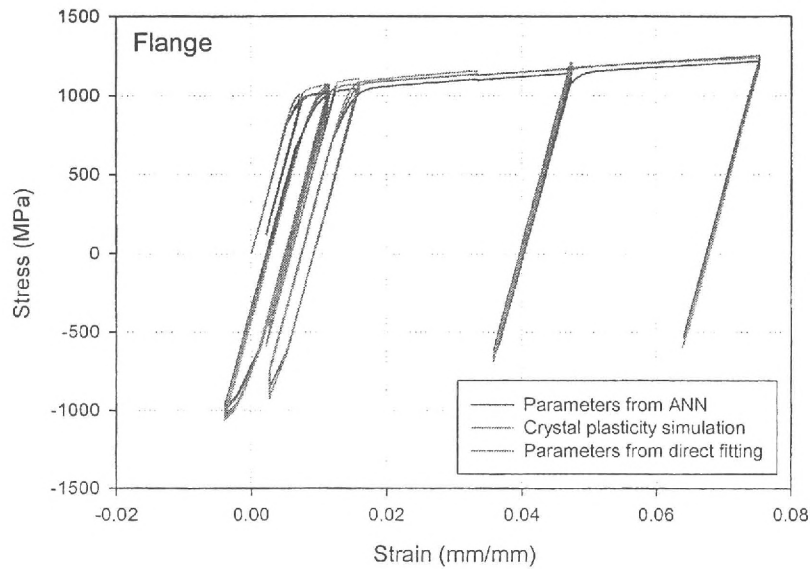


Figure 4. Stress-strain comparison for Flange.

The figures above show that ANN trained macroscale ISV model parameters give a good prediction (comparable to microstructure-sensitive polycrystal plasticity) of the complex cyclic stress-strain response for the microstructures considered.

Task 1. Run a single node to 2% Strain at 1200 °F (650°C)

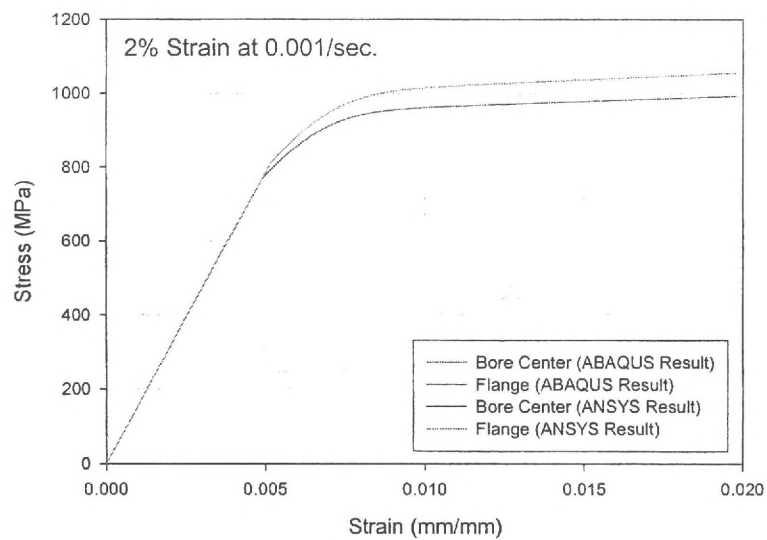


Figure 5. The stress-strain response up to 2% strain at 1200 °F

The simulations were performed using both ABAQUS and ANSYS for comparison of the UMAT implementations in the two different codes. As can be seen in **Figure 5**, ABAQUS and ANSYS implementations give identical results.

Task 2. Strain vs. Time plots (2% - 5% max) for

- 1100 °F / 150 KSI, 160 KSI, 170 KSI
- 1200 °F / 120 KSI, 130 KSI, 140 KSI, 150 KSI, 160 KSI
- 1250 °F / 100 KSI, 120 KSI, 135 KSI
- 1300 °F / 65 KSI, 80 KSI, 100 KSI, 110 KSI, 120 KSI
- 1400 °F / 25 KSI, 40 KSI, 65 KSI

The crystal plasticity model parameters were calibrated using the data obtained from P&W Phase I project. The experimental data were all performed at 1200°F. Hence, only the strain vs. time plots at 1200°F are reliable. **Figure 6** shows the strain versus time responses for various loading conditions over a duration of 120 seconds.

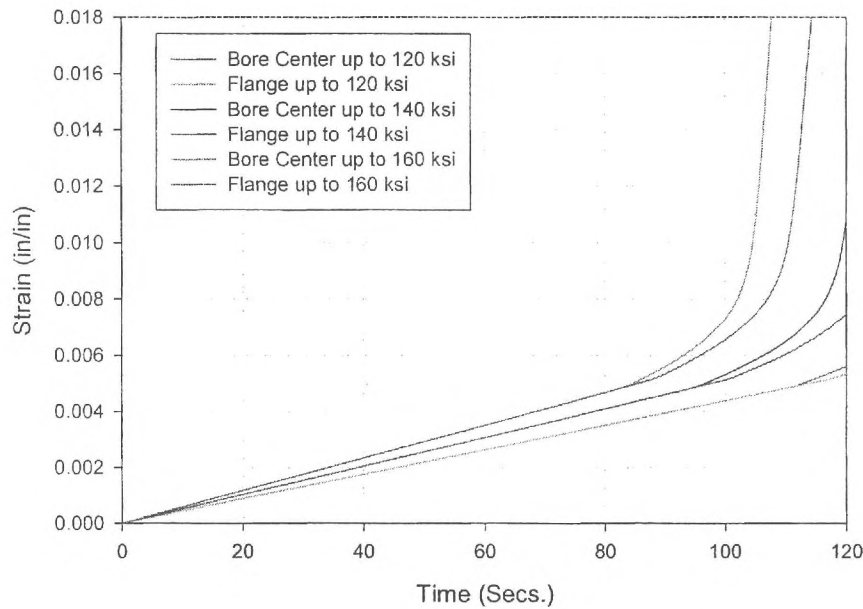


Figure 6. Strain vs time responses for various loading conditions in 120 seconds.

Task 3. Stress vs. Time plots for 5% strain for a simple bar

Initial component testing of the developed Macroscale ISV model for the specified microstructures was performed on a simple bar in 2D with axisymmetric symmetry. The mesh and geometry of the simple bar are given **Figure 7**. **Figures 8** and **9** show the

stress versus time and stress-strain responses, respectively, for the simple bar under completely reversed loading subjected to a maximum strain of 5% which was cycled at 10 cpm.

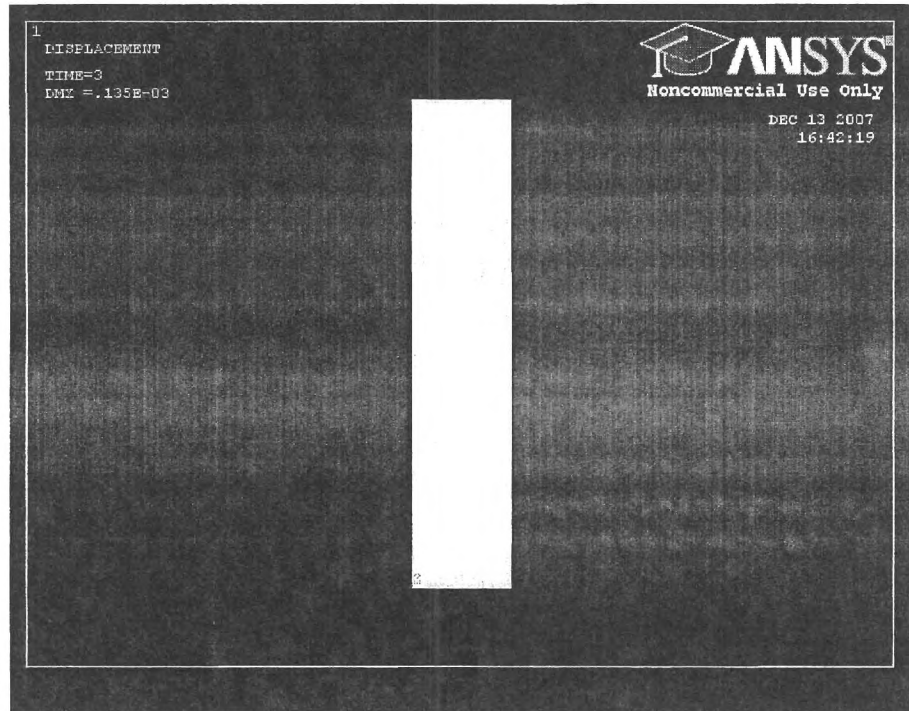


Figure 7: Simple bar modeled by 2D axisymmetric elements in ANSYS.

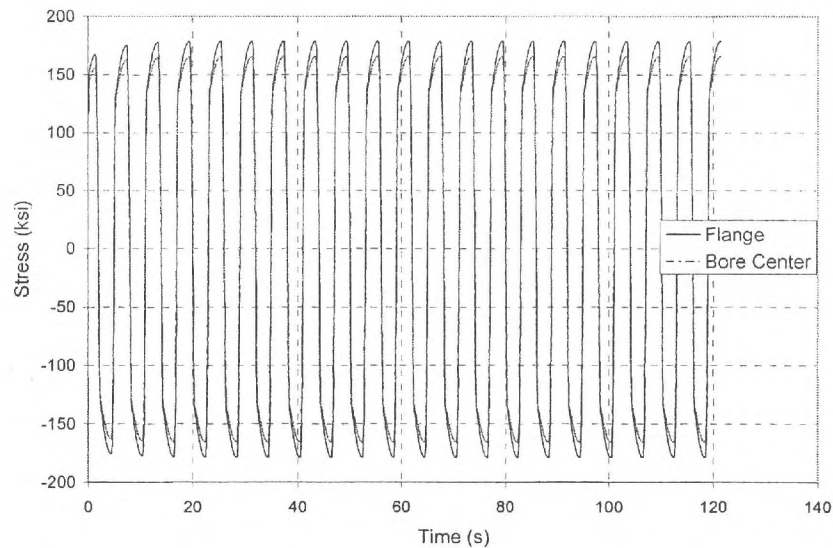


Figure 8: Axial stress versus time for simple bar model with 2D axisymmetric elements for Flange and Bore Center microstructures. The component was cycled in strain control at 10 cpm under completely reversed loading with 5% strain amplitude.

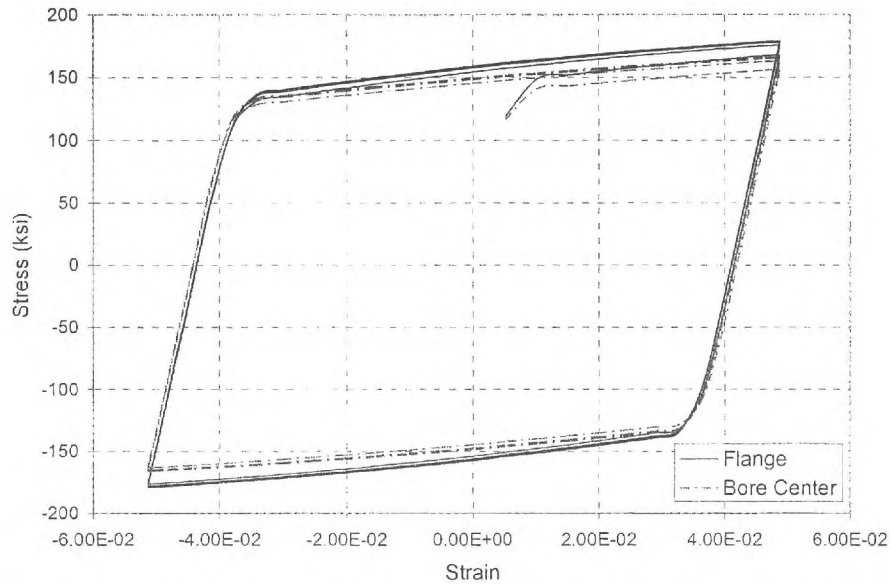


Figure 9: Axial stress versus strain response for simple bar model with 2D axisymmetric elements for Flange and Bore Center microstructures. This component was subjected to 20 cycles in strain control at 10 cpm under completely reversed loading with 5% strain amplitude.

Task 4. Simulations using Pratt & Whitney component.

The component supplied by Pratt & Whitney is shown in **Figure 10** and hereafter will be referred to the PW part. This part is a section of a disk subjected to axisymmetric boundary conditions along both sides to simulate the response of a complete disk even though only a section of the disk is considered. The macroscale ISV model is exercised on the PW part using the material parameters for both the Flange and Bore Center microstructures given in **Table 6** (in English units). In all cases the simulations are performed under stress control and at 10 cpm with stress amplitude of 21253 psi and a stress ratio of 0.05. The maximum stress in the component occurs at the base of the bolt hole. This point is selected as a reference point and is defined specifically as the node of maximum stress under elastic conditions. This reference point is selected by subjecting the PW part to pure linear elastic conditions with the same modulus as the Flange microstructure; in reality any modulus could be used. The Von Mises stress versus time history at the reference point is given in **Figure 11**. **Figure 12** gives the Von Mises stress versus effective strain response at the reference point over 5 cycles. The stress history and stress strain response for the Flange and Bore Center microstructures at the reference point in the direction tangent to the base of the bolt hole for the macroscale ISV model are given in **Figures 13** and **14**. The Von Mises stress profile around the bolt hole in the PW part for the Flange microstructure are given in **Figure 15**.

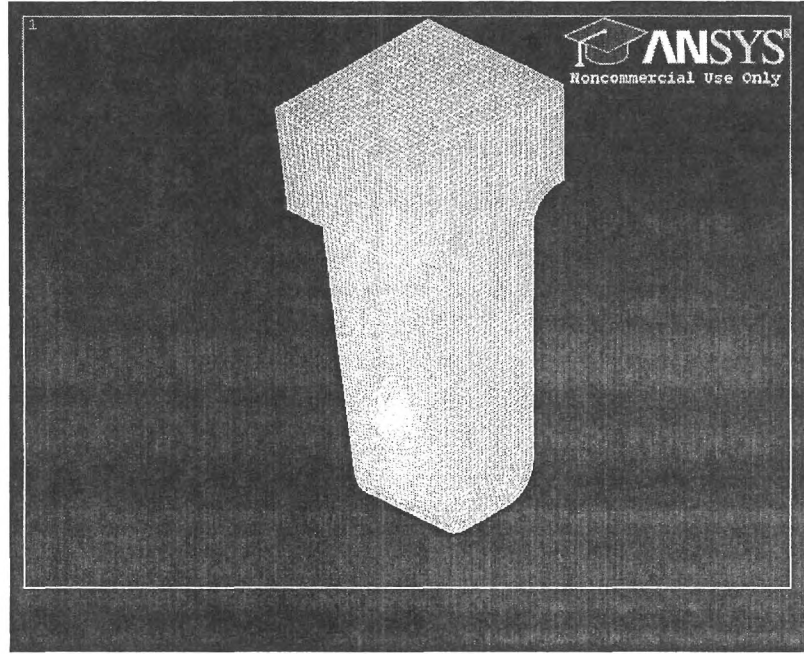


Figure 10. Pratt & Whitney component.

Table 6. Macroscopic viscoplastic model parameters obtained from ANN (650°C) in English units.

Sample	κ (ksi)	A_1 (ksi)	B_1	E_1 (ksi)
"Bore Center"	76.2	29933	1182	64.7
"Flange"	82.4	31648	1179	68.2

Microstructure	E_{mod} (ksi)	ν	A_2 (ksi)	B_2	$A_{v_2} \times 10^{-15}$ (s ⁻¹)
"Bore Center"	22800	0.33	419	7.2	147191
"Flange"	22798	0.33	556	9.6	115612

** A set of parameters were assigned for the model for all microstructures, namely $N = 95$, $N_2 = 7$, $D = 36.3$ ksi, $A_v = 1.25 \times 10^{-3} \text{ s}^{-1}$ and $F_I = 6.5$.

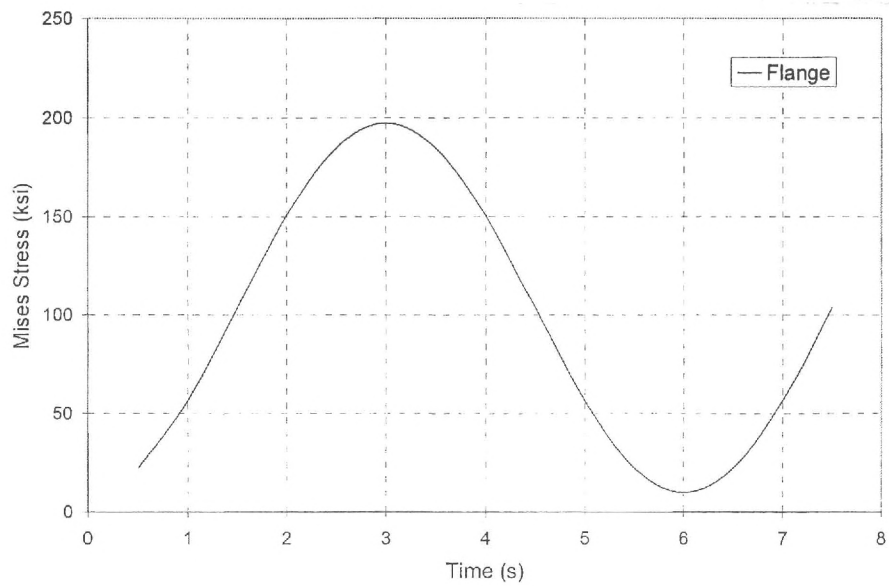


Figure 11. Von Mises stress vs. time response for Flange material under elastic analysis with stress amplitude = 21253 psi; stress ratio = 0.05.

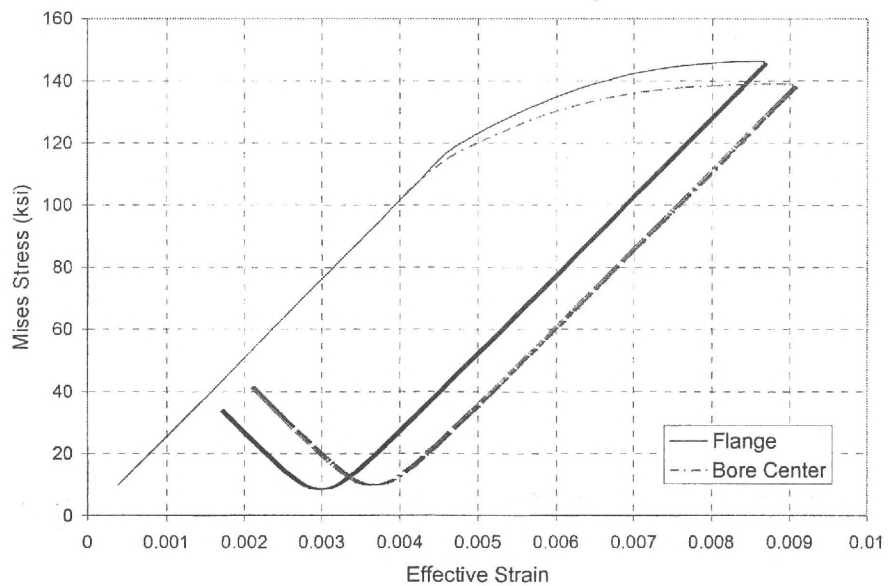


Figure 12. Von Mises stress versus effective strain for Bore Center and Flange.

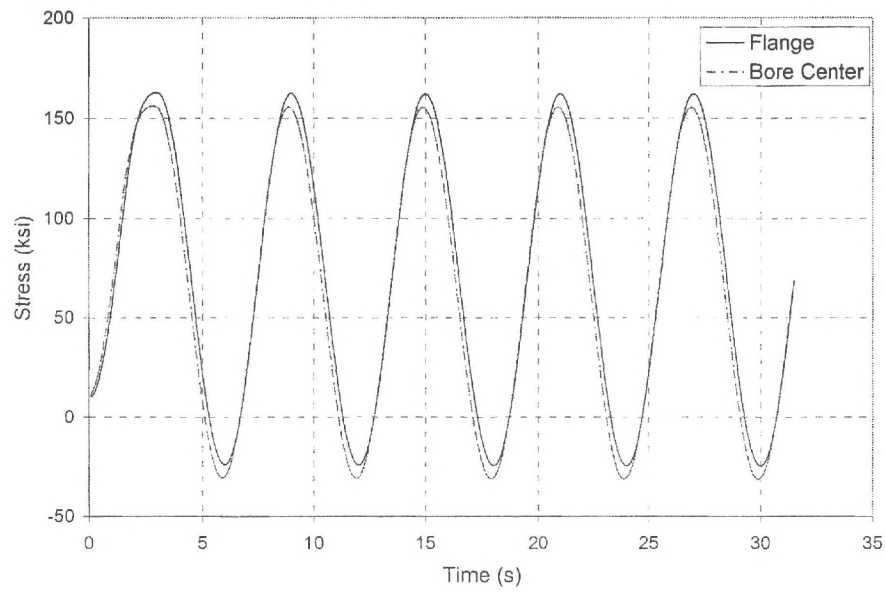


Figure 13. σ_x vs. Time for Bore Center and Flange.

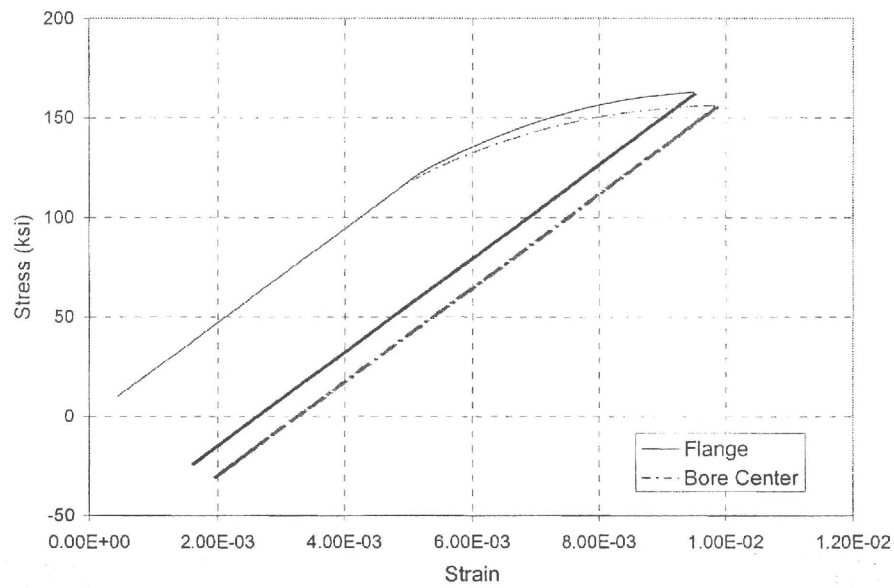


Figure 14. σ_x vs. ε_x for Bore Center and Flange.

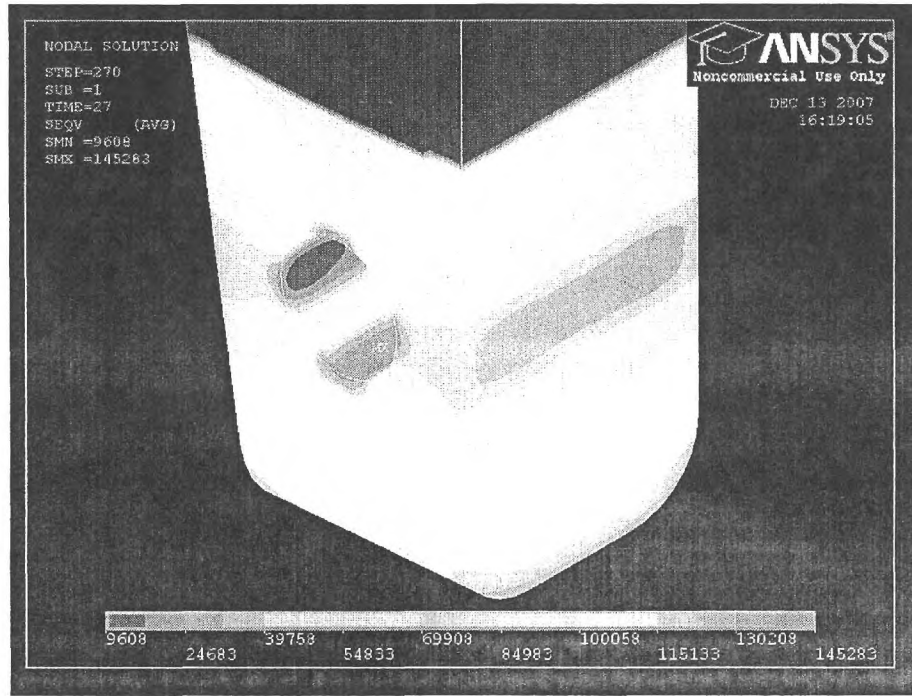


Figure 15. Von Mises stress distribution for Flange microstructure subjected to cyclic elastic analysis with stress amplitude of 21253 psi and a stress ratio of 0.05.

Task 5. Fatigue Indicator Parameters

In polycrystalline materials under multiaxial loading conditions, it has been observed that cracks generally initiate on a critical plane on which the range of cyclic plastic shear strain is maximum under proportional loading (*i.e.*, components of principal stress tensor vary in constant proportion to one another over the stress cycle) [1, 2]. Because of this observation, fatigue indicators have been established relative to the criteria that define these critical planes. An earlier parameter proposed by Smith, Watson and Topper (SWT) considered the coupled effects of stress and strain according to the relationship

$$P_{SWT} = \sqrt{\sigma_{\max} \epsilon_a E} \quad (1)$$

where σ_{\max} is the maximum tensile stress and ϵ_a is the strain amplitude and E is the modulus [3]. The SWT fatigue parameter is intended to capture mainly Stage II crack growth and/or related effects of mean stress for tensile stress-dominated materials, but is not able to account for cracking due to multiaxial loading in shear-dominated materials. Fatemi and Socie [4] (FS) proposed a parameter that accounts for the observation that fatigue cracks initiate on planes of maximum range of plastic shear strain with peak normal stress for a multiaxial cyclic plastic strain state, *i.e.*,

$$P_{FS} = \frac{\Delta \gamma_{p,\max}}{2} \left(1 + k' \frac{\sigma_{n,\max}}{\sigma_y} \right) \quad (2)$$

where $\Delta\gamma_{p,\max}$ is the maximum range of plastic shear strain and $\sigma_{n,\max}$ is the maximum normal stress acting on this plane. Inclusion of the normal stress, $\sigma_{n,\max}$, accounts the effect of normal stress on promoting separation of crack surfaces and is weighted by the material constant k' . This term is normalized by the yield strength. Larger magnitudes of P_{FS} indicate a higher driving force for fatigue damage (crack formation and microstructurally small crack growth). The Fatemi-Socie (FS) parameter has been used to correlate fatigue crack initiation over a large number of grains for a range of multiaxial loading conditions for materials with extended Stage I dominant regions such as 1045 steel and IN718 [2, 5].

The SWT and FS fatigue indicator parameters are calculated for both microstructures at the reference point (i.e., point of maximum stress) in the PW part. **Figure 16** shows the Smith-Watson-Topper parameter over different cycles while **Figure 17** shows the Fatemi-Socie parameter over different cycles. It is expected during shakedown that the cyclic plastic strain amplitude will decrease significantly during the first several cycles as was observed in the decreasing trend of both the SWT and FS parameters. Both microstructures appear to have similar fatigue responses according to these parameters at the reference point. The SWT parameter is estimated to be lower for the Flange microstructure than the Bore Center microstructure; however, when normal stress effects are considered via the FS parameter the results for the two microstructures seem to be much closer.

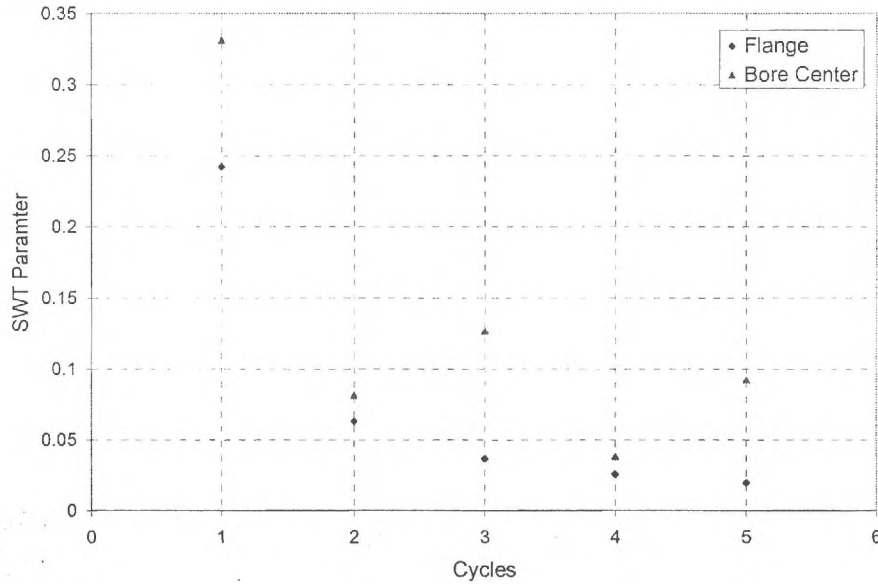


Figure 16. Smith-Watson-Topper parameter over first several cycles.

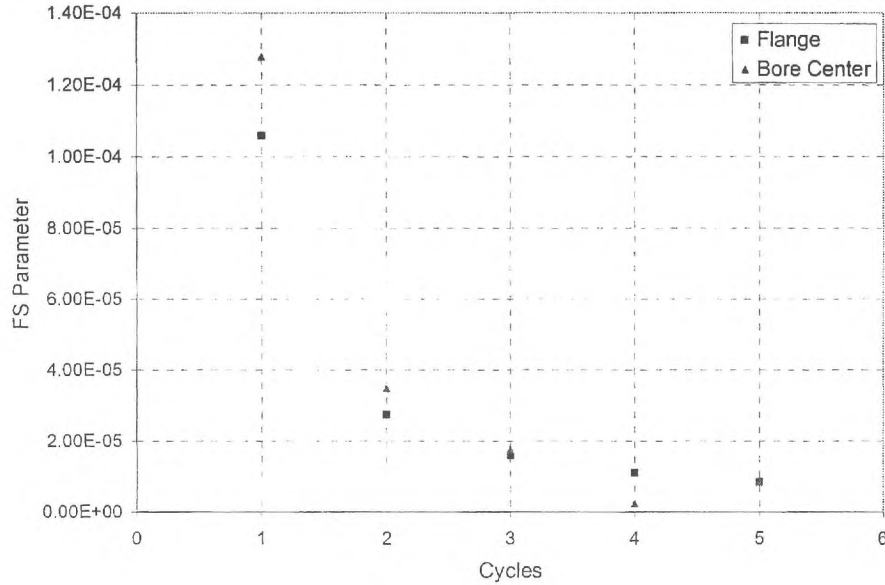


Figure 17. Fatemi-Socie parameter over first several cycles.

5. Development of Ti-6Al-4V Macroscale Model

One strategy for developing a microstructure-sensitive model for Ti-6Al-4V that incorporates effects of crystallographic texture on slip is simply to inform a macroscopic ISV model for each component of texture (random, basal, transverse, basal-transverse) and then weight the multiple models according to the fractions of texture components present in combined textures. Alternatively ANN can be used to perform this task. The crystal plasticity model has been established and validated to serve this purpose.

The challenge in establishing the macroscopic constitutive equations for Ti-6Al-4V lies principally in the fact that the yield surface phase evolves its shape with accumulated inelastic deformation via texture evolution of the low symmetry α phase. Traditional initially isotropic hardening laws cannot accurately model these phenomena. An accurate prediction of the mechanical behaviors of Ti-6Al-4V needs to account for the plastic anisotropy of each phase and must capture the influence of the evolving texture on the inelastic response. The difficulty in formulating the yield surface for textured HCP polycrystals can be attributed to the combination of the strength differential effect (tension vs. compression asymmetry) and the anisotropy (directional dependence of flow stress) due to the texture evolution during the loading process. A model by Cazacu et al. [6], with some enhancements and modifications, is being considered. This model attempts to formulate a macroscopic polycrystal yield surface to account for the strength differential effect associated with low symmetry HCP phase. The form of the yield criterion is given by

$$f(\underline{S}) = (|S_1| - kS_1)^a + (|S_2| - kS_2)^a + (|S_3| - kS_3)^a \quad (3)$$

where S_i , $i=1,\dots,3$ are the principal values of the deviatoric stress tensor. The coefficient k adjusts the strength differential effect, and the integer a is the degree of homogeneity. This isotropic criterion was then extended to the symmetry class of orthotropy (e.g., general rolling texture) by using a linear transformation on the deviatoric stress tensor to give

$$F = (|\Sigma_1| - k\Sigma_1)^a + (|\Sigma_2| - k\Sigma_2)^a + (|\Sigma_3| - k\Sigma_3)^a; \quad \underline{\Sigma} = \underline{\underline{L}} : \underline{S} \quad (4)$$

The restrictions imposed on the fourth-order tensor $\underline{\underline{L}}$ are: (i) to satisfy the major and minor symmetries, and (ii) to be invariant with respect to rotational transformations within the orthotropy group.

Currently, four different textures are being considered in the development of the Ti-6Al-4V macroscale model; they are random, basal, transverse, and basal/transverse textures. The basal plane pole figures of these Ti-6Al-4V textures are given in **Figure 18**.

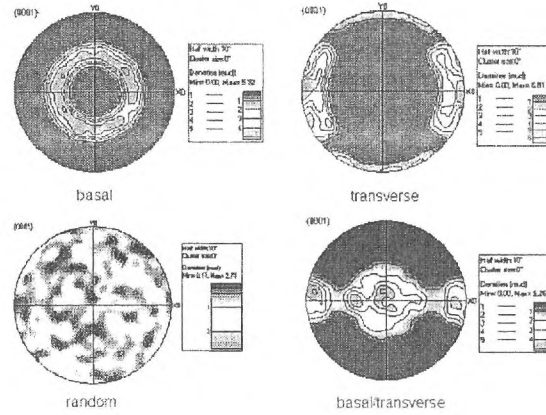


Figure 18. Basal Pole Figures of the common Ti-6Al-4V textures [7].

For each of the texture above, several crystal simulations will be conducted to obtain the yield surface of the material based on the work by Mayeur and McDowell [7] and Zhang et al. [8]. The model equations are summarized in **Table 7**.

Table 7. Equations of the crystal plasticity model for Ti-6Al-4V.

Flow Rule with Back Stress χ^α , Threshold Stress κ^α , and Drag Stress D^α

$$\dot{\gamma}^\alpha = \dot{\gamma}_0 \left\langle \frac{|\tau^\alpha - \chi^\alpha| - \kappa^\alpha}{D^\alpha} \right\rangle^h \text{sgn}(\tau^\alpha - \chi^\alpha)$$

Drag Stress D^α of the primary α -phase is non-evolving, i.e., $\dot{D}^\alpha = 0$

For the HCP slip systems of secondary $\alpha + \beta$ lamellar phase, the Drag Stresses are taken to be the same as their counterparts in the primary α -phase, i.e.

$$D_{\alpha}^{prism} = D_{\alpha+\beta}^{prism}, D_{\alpha}^{basal} = D_{\alpha+\beta}^{basal}$$

Drag stress for the BCC systems is taken to be slightly lower than that for the prismatic systems $D_{\alpha+\beta}^{\{111\}\{110\}} = 0.9 D_{\alpha}^{prism}$

Evolution Equation of the Back Stress:

$$\dot{\chi}^{\alpha} = B\dot{\gamma}^{\alpha} - C\psi(x^{\alpha})\chi^{\alpha}|\dot{\gamma}^{\alpha}| = B\left(\text{sgn}(\dot{\gamma}^{\alpha}) - \psi(x^{\alpha})\frac{C}{B}\chi^{\alpha}\right)|\dot{\gamma}^{\alpha}|$$

$$\psi(x^{\alpha}) = \begin{cases} \exp(\eta_1 x^{\alpha}) & \text{for } x^{\alpha} < 0 \\ 1.0 & \text{for } x^{\alpha} \geq 0 \end{cases}$$

$$x^{\alpha} = \text{sgn}(\tau^{\alpha} - \chi^{\alpha})p^{\alpha}, \text{ with } \dot{p}^{\alpha} = |\dot{\gamma}^{\alpha}|\text{sgn}(\tau^{\alpha} - \chi^{\alpha})$$

Evolution Equation of the Threshold Stress:

- For the pyramidal slip systems of the primary α -phase and the hard slip system of the $\alpha + \beta$ colony, only one term is in effect, i.e. $\kappa_1^{\alpha} = k_y(d^{\alpha})^{-0.5}$
- For the basal and prismatic slip systems of the primary α -phase, the evolution equation includes additional terms so as to give

$$\kappa_{prism,basal}^{\alpha} = \kappa_1^{\alpha} + A(\tau_{pyr1} - \tau_{pyr2}) + \kappa_s^{\alpha}|_{prism,basal}$$

$$\dot{\kappa}_s^{\alpha}|_{prism,basal} = -\mu\kappa_s^{\alpha}|_{prism,basal}\|\dot{\gamma}^{\alpha}\| \quad \text{with } \kappa_s^{\alpha}|_{prism,basal}(0) = \kappa_s(0)$$

- The Threshold stress for the soft slip systems of the $\alpha + \beta$ colony is given by

$$\kappa_{soft}^{\alpha} = \kappa_1^{\alpha} + \kappa_s^{\alpha}|_{soft}$$

$$\dot{\kappa}_s^{\alpha}|_{soft} = -\mu\kappa_s^{\alpha}|_{soft}\|\dot{\gamma}^{\alpha}\|$$

$$\kappa_s^{\alpha}|_{soft}(0) = \kappa_s(0)$$

Additional steps to achieve microstructure-sensitive macroscopic model for Ti-6Al-4V:

- Conduct crystal plasticity runs of Ti-6Al-4V for key texture components.
- Fit the yield surface to the form suggested by Cazacu or comparable form to fit crystal plasticity simulations.
- Formulate the elements of viscoplastic constitutive model with texture representation (back stress)
- Conduct ANN to link the Ti-6Al-4V crystal plasticity model to the microstructure-sensitive macroscale viscoplastic model.

References

- [1] D. L. McDowell, "Multiaxial Fatigue Strength," in *ASM Handbook*. Vol. 19 Materials Park, OH: ASM International, 1996, pp. 263-273.
- [2] D. F. Socie, "Critical Plane Approaches for Multiaxial Fatigue Damage Assessment," in *Advances in Multiaxial Fatigue, ASTM STP 1191*, D. L. McDowell and R. Ellis, Eds. Philadelphia: American Society of Testing and Materials, 1993, pp. 7-36.
- [3] K. N. Smith, P. Watson, and T. H. Topper, "A Stress-Strain Function for the Fatigue of Metals," *Journal of Materials*, Vol. 5, 1970, pp. 767-778.
- [4] A. Fatemi and D. F. Socie, "A critical plane approach to multiaxial fatigue damage including out-of-phase loading," *Fatigue and Fracture of Engineering Materials and Structures*, Vol. 11, 1988, pp. 149-65.
- [5] D. L. McDowell and J. Y. Berard, "A ΔJ -based approach to biaxial fatigue," *Fatigue & Fracture of Engineering Materials & Structures*, Vol. 15, 1992, pp. 719-41.
- [6] O. Cazacu, B. Plunkett, and F. Barlat, "Orthotropic yield criterion for hexagonal closed packed metals," *International Journal of Plasticity*, Vol. 22, 2006, pp. 1171-1194.
- [7] J. R. Mayeur and D. L. McDowell, "A Three-Dimensional Crystal Plasticity Model for Duplex Ti-6Al-4V," *International Journal of Plasticity*, Vol. 23, No. 9, 2007, pp. 1457-1485.
- [8] M. Zhang, J. Zhang, and D. L. McDowell, "Microstructure-Based Crystal Plasticity Modeling of Cyclic Deformation of Ti-6Al-4V," *International Journal of Plasticity*, Vol. 23, No. 8, 2007, pp. 1328-1348.

Appendix A. Crystal Plasticity Constitutive Equations

Microstructure-Dependent Crystal Plasticity Model for a Polycrystalline Nickel-Base Superalloy

Flow Rule

$$\dot{\gamma}^\alpha = \left[\dot{\gamma}_o \left\langle \frac{|\tau^\alpha - \chi^\alpha| - \kappa_\lambda^\alpha}{D^\alpha} \right\rangle^{n_1} + \dot{\gamma}_1 \left\langle \frac{|\tau^\alpha - \chi^\alpha|}{D^\alpha} \right\rangle^{n_2} \right] \text{sgn}(\tau^\alpha - \chi^\alpha) \quad (1)$$

where

$$\begin{aligned} \kappa_\lambda^\alpha &= \kappa_{o,\lambda}^\alpha + \alpha \mu_{\text{mix}} b \sqrt{\rho_\lambda^\alpha}, \quad \lambda = \text{oct, cub} \\ \kappa_{o,\text{oct}}^\alpha &= \left[(\tau_{o,\lambda}^\alpha)^{n_\kappa} + \Psi_{\text{oct}}(f_{p_1}, d_2, f_{p_2}, d_3, f_{p_3})^{n_\kappa} \right]^{1/n_\kappa} + \frac{(f'_{p_1} + f'_{p_2})}{f_{p_{\text{ref}}}} \tau_{ns}^\alpha \\ \kappa_{o,\text{cub}}^\alpha &= \left[(\tau_{o,\lambda}^\alpha)^{n_\kappa} + \Psi_{\text{cub}}(f_{p_1}, d_2, f_{p_2}, d_3, f_{p_3})^{n_\kappa} \right]^{1/n_\kappa} \\ \tau_{ns}^\alpha &= h_{pe} \tau_{pe}^\alpha + h_{cb} |\tau_{cb}^\alpha| + h_{se} \tau_{se}^\alpha \\ \Psi_{\text{oct}} &= \Psi_{\text{cub}} = c_{p_1} \sqrt{0.81 w f'_{p_1} \mu_{\gamma'} \frac{\Gamma_{\text{APB}}}{d_1}} + c_{p_2} \sqrt{0.81 w f'_{p_2} \mu_{\gamma'} \frac{\Gamma_{\text{APB}}}{d_2}} \\ &\quad + c_{p_3} \sqrt{\mu_{\gamma'} \frac{d_3}{d_{3,\text{ref}}} f'_{p_3} \frac{\Gamma_{\text{APB}}}{b}} + c_{gr} d_{gr}^{-0.5} \\ f'_{p_1} &= \frac{f_{p_1}}{f_{p_1} + f_m}, \quad f'_{p_2} = \frac{f_{p_2}}{f_{p_2} + f_m}, \quad f'_{p_3} = \frac{f_{p_3}}{f_{p_3} + f_m} \end{aligned}$$

Internal State Variables

a). Dislocation Density

$$\begin{aligned} \dot{\rho}_\lambda^{(\alpha)} &= \sum_\beta h^{(\alpha\beta)} \left\{ Z_o + k_1 \sqrt{\rho_\lambda^{(\beta)}} - k_2 \rho_\lambda^{(\beta)} \right\} |\dot{\gamma}^{(\beta)}|, \quad h^{(\alpha\beta)} = 1 \\ Z_o &= \frac{k_\delta}{b d_{\delta_{\text{eff}}}}, \quad d_{\delta_{\text{eff}}} = \left(\frac{1}{d_{1\delta}} + \frac{1}{d_{2\delta}} \right)^{-1} \end{aligned} \quad (2)$$

b). *Backstress*

$$\begin{aligned}\dot{\chi}_{\lambda}^{(\alpha)} &= C_{\chi} \left\{ \eta \mu_s b \sqrt{\rho_{\lambda}^{(\alpha)}} \operatorname{sgn} \left(\tau^{(\alpha)} - \chi_{\lambda}^{(\alpha)} \right) - \chi_{\lambda}^{(\alpha)} \right\} |\dot{\gamma}^{(\alpha)}| \\ \eta &= \frac{\eta_o Z_o}{Z_o + k_1 \sqrt{\rho_{\lambda}^{(\alpha)}}}\end{aligned}\tag{3}$$

Appendix B. Walker – Type Constitutive Equations

Uniaxial Walker Type UMAT (McDowell Form)

Two-Term Flow Rule and Two Backstress Formulation

Flow Rule

$$\begin{aligned} \dot{\epsilon}^n = D^n = & \left\{ A_v \left\langle \frac{\sigma^v - \kappa - K_{soft} - R}{D_1} \right\rangle^N \exp \left[B_v \left\langle \frac{\sigma^v - \kappa - K_{soft} - R}{D_1} \right\rangle^{N+1} \right] \right. \\ & \left. + A_{v_2} \left\langle \frac{\sigma^v}{D_2} \right\rangle^{N_2} \exp \left[B_{v_2} \left\langle \frac{\sigma^v}{D_2} \right\rangle^{N_2+1} \right] \right\} \text{sgn}(\sigma - \Omega), \quad \sigma^v = |\sigma - \Omega| \end{aligned} \quad (1)$$

Hardening Evolution

$$K_{soft} = \sum_{i=1}^2 K_1^i \exp \left\{ -m_k \frac{\Omega_{\max}^i}{\Omega_{\lim}^i} \right\}, \quad \Omega_{\max}^i = \max |\Omega^i| \quad (2)$$

$$R = Q_h \{1 - \exp(-b_h p)\}, \quad p = \int |\dot{\epsilon}^n| dt \quad (3)$$

$$\dot{\Omega} = \sum_{i=1}^2 \left[\frac{A^i}{\Omega_{\lim}^i} \left(\Omega_{\lim}^i \dot{\epsilon}^n - g \left(\frac{\Omega^i}{\Omega_{\lim}^i} \right) \Omega^i |\dot{\epsilon}^n| \right) \right] \quad (4)$$

$$g \left(\frac{\Omega^i}{\Omega_{\lim}^i} \right) = \left\langle \frac{\Omega^i}{|\Omega^i|} \frac{\dot{\epsilon}^n}{|\dot{\epsilon}^n|} \right\rangle^{\chi_i+1} \left(\frac{|\Omega^i|}{\Omega_{\lim}^i} \right)^{m_g^i}$$

Note: The uniaxial backstress Ω is assumed to be deviatoric. K_{soft} term was neglected since there was no softening observed. This term had originally been placed in the denominator, but was then placed in the numerator.

Appendix C. Walker – Type UMAT

Walker Type UMAT (McDowell Form)

Two-Term Flow Rule and Two Backstress Formulation

1 Constitutive Equations

Flow Rule

$$\begin{aligned} \dot{\epsilon}^n = D^n = \dot{p}n = & \left\{ A_v \sqrt{\frac{3}{2}} \left(\frac{\langle S_v \rangle}{D} \right)^N \exp \left[B_v \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right] \right. \\ & \left. + A_{v_2} \sqrt{\frac{3}{2}} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2} \exp \left[B_{v_2} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right] \right\} n \end{aligned} \quad (1)$$

where

$$S_v = \sqrt{\frac{3}{2}} \|\sigma' - \alpha\| - \kappa - R, \quad S_{v_2} = \sqrt{\frac{3}{2}} \|\sigma' - \alpha\|$$

$$n = \frac{\sigma' - \alpha}{\|\sigma' - \alpha\|}, \quad \|\sigma' - \alpha\| = [(\sigma' - \alpha) : (\sigma' - \alpha)]^{1/2}$$

Hardening Evolution

$$\dot{\alpha} = \sum_{i=1}^2 \left(\frac{2}{3} A_i n - \sqrt{\frac{2}{3}} B_i \alpha_i \right) \dot{p}, \quad \dot{R} = (E_1 - F_1 R) \sqrt{\frac{2}{3}} \dot{p}$$

Back to (1),

$$\dot{p} = A_v \sqrt{\frac{3}{2}} \left\{ \left(\frac{\langle S_v \rangle}{D} \right)^N \exp \left[B_v \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right] \right\} + A_{v_2} \sqrt{\frac{3}{2}} \left\{ \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2} \exp \left[B_{v_2} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right] \right\}$$

$$g = \Delta p - A_v \sqrt{\frac{3}{2}} \left\{ \left(\frac{\langle S_v \rangle}{D} \right)^N \exp \left[B_v \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right] \right\} \Delta t$$

$$- A_{v_2} \sqrt{\frac{3}{2}} \left\{ \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2} \exp \left[B_{v_2} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right] \right\} \Delta t$$

$$g_i = \Delta p_i - \left[A_v \sqrt{\frac{3}{2}} \left\{ \left(\frac{\langle S_v \rangle}{D} \right)^N \exp \left[B_v \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right] \right\} \right]_i \Delta t$$

$$- \left[A_{v_2} \sqrt{\frac{3}{2}} \left\{ \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2} \exp \left[B_{v_2} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right] \right\} \right]_i \Delta t$$

1.1 Newton-Raphson Scheme

The next increment, $g_{i+1} = 0$, is given by

$$0 = g_{i+1} = g_i + \frac{\partial g}{\partial \Delta p} \Big|_i \Delta(\Delta p) = g_i + \frac{\partial g}{\partial \Delta p} \Big|_i (\Delta p_{i+1} - \Delta p_i)$$

$$\Delta p_{i+1} = \Delta p_i - \frac{g_i}{[\partial g / \partial \Delta p]_i}$$

where

$$g = \Delta p - A_v \sqrt{\frac{3}{2}} \Delta t F(\sigma', \alpha, R) - A_{v_2} \sqrt{\frac{3}{2}} \Delta t F_2(\sigma', \alpha)$$

$$F(\sigma', \alpha, R) = \left(\frac{\langle S_v \rangle}{D} \right)^N \exp \left[B_v \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right], \quad F_2(\sigma', \alpha) = \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2} \exp \left[B_{v_2} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right]$$

$$\frac{\partial g}{\partial \Delta p} = 1 - A_v \sqrt{\frac{3}{2}} \Delta t \left(\frac{\partial F}{\partial \sigma'} : \frac{\partial \sigma'}{\partial \Delta p} + \frac{\partial F}{\partial \alpha} : \frac{\partial \alpha}{\partial \Delta p} + \frac{\partial F}{\partial R} \frac{\partial R}{\partial \Delta p} \right)$$

$$- A_{v_2} \sqrt{\frac{3}{2}} \Delta t \left(\frac{\partial F_2}{\partial \sigma'} : \frac{\partial \sigma'}{\partial \Delta p} + \frac{\partial F_2}{\partial \alpha} : \frac{\partial \alpha}{\partial \Delta p} \right) \quad (2)$$

Now, we need to calculate the components from (2).

Recalling F ,

$$F = \left(\frac{\langle S_v \rangle}{D} \right)^N \exp \left[B_v \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right]$$

noting that

$$S_v = \sqrt{\frac{3}{2}} [(\sigma' - \alpha) : (\sigma' - \alpha)]^{1/2} - \kappa - R$$

So,

$$\begin{aligned}
\frac{\partial F}{\partial \sigma'} &= \left\{ \frac{N}{D} \left(\frac{\langle S_v \rangle}{D} \right)^{N-1} \exp \left[B_v \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right] \right. \\
&\quad \left. + \left(\frac{\langle S_v \rangle}{D} \right)^N \frac{B_v}{D} (N+1) \left(\frac{\langle S_v \rangle}{D} \right)^N \exp \left[B_v \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right] \right\} \frac{\partial S_v}{\partial \sigma'} \\
&= \left\{ \frac{N}{D} \left(\frac{\langle S_v \rangle}{D} \right)^{N-1} + \frac{B_v}{D} (N+1) \left(\frac{\langle S_v \rangle}{D} \right)^{2N} \right\} \exp \left[B_v \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right] \frac{\partial S_v}{\partial \sigma'} \\
&= \left[N + B_v (N+1) \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right] \frac{1}{D} \left(\frac{\langle S_v \rangle}{D} \right)^{N-1} \exp \left[B_v \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right] \frac{\partial S_v}{\partial \sigma'}
\end{aligned}$$

Next,

$$\begin{aligned}
\frac{\partial S_v}{\partial \sigma'} &= \frac{\partial}{\partial \sigma'} \left(\sqrt{\frac{3}{2}} [(\sigma' - \alpha) : (\sigma' - \alpha)]^{1/2} - R \right) = \sqrt{\frac{3}{2}} \frac{\partial}{\partial \sigma'} \left([(\sigma' - \alpha) : (\sigma' - \alpha)]^{1/2} \right) \\
&= \sqrt{\frac{3}{2}} \left(\frac{1}{2} \frac{2(\sigma' - \alpha)}{[(\sigma' - \alpha) : (\sigma' - \alpha)]^{1/2}} \right) \frac{\partial (\sigma' - \alpha)}{\partial \sigma'} = \sqrt{\frac{3}{2}} \left(\frac{(\sigma' - \alpha)}{[(\sigma' - \alpha) : (\sigma' - \alpha)]^{1/2}} \right) \\
&= \sqrt{\frac{3}{2}} n, \quad \text{noting } n = \frac{(\sigma' - \alpha)}{[(\sigma' - \alpha) : (\sigma' - \alpha)]^{1/2}}
\end{aligned}$$

Finally

$$\frac{\partial F}{\partial \sigma'} = \sqrt{\frac{3}{2}} n \left[N + B_v (N+1) \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right] \frac{1}{D} \left(\frac{\langle S_v \rangle}{D} \right)^{N-1} \exp \left[B_v \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right]$$

We can write,

$$\frac{\partial F}{\partial \sigma'} = \frac{n F_v}{A_v \sqrt{\frac{3}{2}} \Delta t} = -\frac{\partial F}{\partial \alpha}, \quad \frac{\partial F}{\partial R} = \frac{-F_v}{\frac{3 A_v}{2} \Delta t}$$

where

$$F_v = \frac{3}{2} A_v \Delta t \left[N + B_v (N+1) \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right] \frac{1}{D} \left(\frac{\langle S_v \rangle}{D} \right)^{N-1} \exp \left[B_v \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right]$$

Note that

$$\frac{\partial \sigma'}{\partial \Delta p} = -2\mu n, \quad \frac{\partial \alpha}{\partial \Delta p} = \frac{2}{3} (A_1 + A_2) n - \sqrt{\frac{2}{3}} (B_1 \alpha_1 + B_2 \alpha_2), \quad \frac{\partial R}{\partial \Delta p} = \sqrt{\frac{2}{3}} (E_1 - F_1 R)$$

Now, work on the second term of flow rule

$$\begin{aligned}
\frac{\partial F_2}{\partial \sigma'} &= \left\{ \frac{N_2}{D} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2-1} \exp \left[B_{v_2} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right] \right. \\
&\quad \left. + \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2} \frac{B_{v_2}}{D} (N_2 + 1) \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2} \exp \left[B_{v_2} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right] \right\} \frac{\partial S_{v_2}}{\partial \sigma'} \\
&= \left\{ \frac{N_2}{D} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2-1} + \frac{B_{v_2}}{D} (N_2 + 1) \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{2N_2} \right\} \exp \left[B_{v_2} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right] \frac{\partial S_{v_2}}{\partial \sigma'} \\
&= \left[N_2 + B_{v_2} (N_2 + 1) \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right] \frac{1}{D} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2-1} \exp \left[B_{v_2} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right] \frac{\partial S_{v_2}}{\partial \sigma'}
\end{aligned}$$

Next,

$$\begin{aligned}
\frac{\partial S_{v_2}}{\partial \sigma'} &= \frac{\partial}{\partial \sigma'} \left(\sqrt{\frac{3}{2}} [(\sigma' - \alpha) : (\sigma' - \alpha)]^{1/2} \right) = \sqrt{\frac{3}{2}} \frac{\partial}{\partial \sigma'} \left([(\sigma' - \alpha) : (\sigma' - \alpha)]^{1/2} \right) \\
&= \sqrt{\frac{3}{2}} \left(\frac{1}{2} \frac{2(\sigma' - \alpha)}{[(\sigma' - \alpha) : (\sigma' - \alpha)]^{1/2}} \right) \frac{\partial (\sigma' - \alpha)}{\partial \sigma'} = \sqrt{\frac{3}{2}} \left(\frac{(\sigma' - \alpha)}{[(\sigma' - \alpha) : (\sigma' - \alpha)]^{1/2}} \right) \\
&= \sqrt{\frac{3}{2}} \mathbf{n}, \quad \text{noting } \mathbf{n} = \frac{(\sigma' - \alpha)}{[(\sigma' - \alpha) : (\sigma' - \alpha)]^{1/2}}
\end{aligned}$$

Finally, for the second term of flow rule

$$\frac{\partial F_2}{\partial \sigma'} = \sqrt{\frac{3}{2}} \mathbf{n} \left[N_2 + B_{v_2} (N_2 + 1) \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right] \frac{1}{D} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2-1} \exp \left[B_{v_2} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right]$$

We can write

$$\frac{\partial F_2}{\partial \sigma'} = \frac{\mathbf{n} F_{v_2}}{A_{v_2} \sqrt{\frac{3}{2}} \Delta t} = - \frac{\partial F_2}{\partial \alpha}$$

where

$$F_{v_2} = \frac{3}{2} A_{v_2} \Delta t \left[N_2 + B_{v_2} (N_2 + 1) \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right] \frac{1}{D} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2-1} \exp \left[B_{v_2} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right]$$

Note that

$$\frac{\partial \sigma'}{\partial \Delta p} = -2\mu \mathbf{n}, \quad \frac{\partial \alpha}{\partial \Delta p} = \frac{2}{3} (A_1 + A_2) \mathbf{n} - \sqrt{\frac{2}{3}} (B_1 \alpha_1 + B_2 \alpha_2)$$

Henceforth,

$$\begin{aligned}
\frac{\partial g}{\partial \Delta p} &= 1 - A_v \sqrt{\frac{3}{2}} \Delta t \left(\frac{n F_v}{A_v \sqrt{\frac{3}{2}} \Delta t} : (-2\mu n) - \frac{n F_v}{A_v \sqrt{\frac{3}{2}} \Delta t} : \left(\frac{2}{3} (A_1 + A_2) n - \sqrt{\frac{2}{3}} (B_1 \alpha_1 + B_2 \alpha_2) \right) \right. \\
&\quad \left. - \frac{F_v}{\frac{3A_v}{2} \Delta t} \sqrt{\frac{2}{3}} (E_1 - F_1 R) \right) - A_{v_2} \sqrt{\frac{3}{2}} \Delta t \left(\frac{n F_{v_2}}{A_{v_2} \sqrt{\frac{3}{2}} \Delta t} : (-2\mu n) \right. \\
&\quad \left. - \frac{n F_{v_2}}{A_{v_2} \sqrt{\frac{3}{2}} \Delta t} : \left(\frac{2}{3} (A_1 + A_2) n - \sqrt{\frac{2}{3}} (B_1 \alpha_1 + B_2 \alpha_2) \right) \right) \\
&= 1 + F_v \left(n : \left(2\mu n + \frac{2}{3} (A_1 + A_2) n - \sqrt{\frac{2}{3}} (B_1 \alpha_1 + B_2 \alpha_2) \right) + \frac{2}{3} (E_1 - F_1 R) \right) \\
&\quad + F_{v_2} \left(n : \left(2\mu n + \frac{2}{3} (A_1 + A_2) n - \sqrt{\frac{2}{3}} (B_1 \alpha_1 + B_2 \alpha_2) \right) \right) \tag{3}
\end{aligned}$$

1.2 Calculating the Jacobian

The Jacobian, \mathbf{J} , is given by

$$\begin{aligned}
\mathbf{J} &= \frac{\partial \Delta \boldsymbol{\sigma}}{\partial \Delta \boldsymbol{\varepsilon}} = \frac{\partial}{\partial \Delta \boldsymbol{\varepsilon}} [\mathbf{C} : (\mathbf{D} - \mathbf{D}^n) \Delta t] = \frac{\partial}{\partial \Delta \boldsymbol{\varepsilon}} [(\mathbf{C} : \mathbf{D}) \Delta t - (\mathbf{C} : \mathbf{D}^n) \Delta t] \\
&= \frac{\partial}{\partial \Delta \boldsymbol{\varepsilon}} [(\mathbf{C} : \Delta \boldsymbol{\varepsilon}) - (\mathbf{C} : \Delta p \mathbf{n})], \text{ recall } \mathbf{D} = \dot{\boldsymbol{\varepsilon}}, \quad \mathbf{D}^n \Delta t = \Delta p \mathbf{n} \\
&= \mathbf{C} : \left(\mathbf{I} - \mathbf{n} \frac{\partial \Delta p}{\partial \Delta \boldsymbol{\varepsilon}} \right) = \mathbf{C} - (\mathbf{C} : \mathbf{n}) \otimes \frac{\partial \Delta p}{\partial \Delta \boldsymbol{\varepsilon}} \tag{4}
\end{aligned}$$

Since,

$$\dot{p} = \left\{ A_v \sqrt{\frac{3}{2}} \left(\frac{\langle S_v \rangle}{D} \right)^N \exp \left[B_v \left(\frac{\langle S_v \rangle}{D} \right)^{N+1} \right] \right\} + \left\{ A_{v_2} \sqrt{\frac{3}{2}} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2} \exp \left[B_{v_2} \left(\frac{\langle S_{v_2} \rangle}{D} \right)^{N_2+1} \right] \right\}$$

$$\Delta p = A_v \sqrt{\frac{3}{2}} \Delta t F + A_{v_2} \sqrt{\frac{3}{2}} \Delta t F_2$$

then,

$$\begin{aligned}\frac{\partial \Delta p}{\partial \Delta \epsilon} &= A_v \sqrt{\frac{3}{2}} \Delta t \left(\frac{\partial F}{\partial \sigma'} : \frac{\partial \sigma'}{\partial \Delta \epsilon} + \frac{\partial F}{\partial \alpha} : \frac{\partial \alpha}{\partial \Delta p} \otimes \frac{\partial \Delta p}{\partial \Delta \epsilon} + \frac{\partial F}{\partial R} \frac{\partial R}{\partial \Delta p} \otimes \frac{\partial \Delta p}{\partial \Delta \epsilon} \right) \\ &\quad + A_{v_2} \sqrt{\frac{3}{2}} \Delta t \left(\frac{\partial F_2}{\partial \sigma'} : \frac{\partial \sigma'}{\partial \Delta \epsilon} + \frac{\partial F_2}{\partial \alpha} : \frac{\partial \alpha}{\partial \Delta p} \otimes \frac{\partial \Delta p}{\partial \Delta \epsilon} \right)\end{aligned}$$

We already know that

$$\frac{\partial F}{\partial \sigma'} = \frac{n F_v}{A_v \sqrt{\frac{3}{2}} \Delta t} = -\frac{\partial F}{\partial \alpha}, \quad \frac{\partial F}{\partial R} = \frac{-F_v}{\frac{3 A_v}{2} \Delta t}, \quad \frac{\partial F_2}{\partial \sigma'} = \frac{n F_{v_2}}{A_{v_2} \sqrt{\frac{3}{2}} \Delta t} = -\frac{\partial F_2}{\partial \alpha}$$

We know that

$$\sigma' = \sigma - \frac{1}{3} \sigma_{kk} \mathbf{I} = \sigma - K \varepsilon_{kk} \mathbf{I}, \quad \frac{\partial \Delta \sigma'}{\partial \Delta \epsilon} = \frac{\partial \Delta \sigma}{\partial \Delta \epsilon} - \frac{\partial}{\partial \Delta \epsilon} (K \Delta \varepsilon_{kk} \mathbf{I})$$

Assuming that K is a constant during deformation

$$\frac{\partial \Delta \sigma'}{\partial \Delta \epsilon} = \frac{\partial \Delta \sigma}{\partial \Delta \epsilon} - K (\mathbf{I} \otimes \mathbf{I}) = \mathbf{C} - (\mathbf{C} : \mathbf{n}) \otimes \frac{\partial \Delta p}{\partial \Delta \epsilon} - K (\mathbf{I} \otimes \mathbf{I})$$

$$\frac{\partial \alpha}{\partial \Delta p} = \frac{2}{3} (A_1 + A_2) \mathbf{n} - \sqrt{\frac{2}{3}} (B_1 \alpha_1 + B_2 \alpha_2), \quad \frac{\partial R}{\partial \Delta p} = \sqrt{\frac{2}{3}} (E_1 - F_1 R)$$

So now,

$$\begin{aligned}\frac{\partial \Delta p}{\partial \Delta \epsilon} &= A_v \sqrt{\frac{3}{2}} \Delta t \left[\left(\frac{n F_v}{A_v \sqrt{\frac{3}{2}} \Delta t} \right) : \left(\mathbf{C} - (\mathbf{C} : \mathbf{n}) \otimes \frac{\partial \Delta p}{\partial \Delta \epsilon} - K (\mathbf{I} \otimes \mathbf{I}) \right) \right. \\ &\quad + \left(\frac{-n F_v}{A_v \sqrt{\frac{3}{2}} \Delta t} \right) : \left(\frac{2}{3} (A_1 + A_2) \mathbf{n} - \sqrt{\frac{2}{3}} (B_1 \alpha_1 + B_2 \alpha_2) \right) \otimes \frac{\partial \Delta p}{\partial \Delta \epsilon} \\ &\quad \left. + \left(\frac{-F_v}{\frac{3 A_v}{2} \Delta t} \right) \sqrt{\frac{2}{3}} (E_1 - F_1 R) \otimes \frac{\partial \Delta p}{\partial \Delta \epsilon} \right]\end{aligned}$$

$$\begin{aligned}
& + A_{v_2} \sqrt{\frac{3}{2}} \Delta t \left[\left(\frac{n F_{v_2}}{A_{v_2} \sqrt{\frac{3}{2}} \Delta t} \right) : \left(\mathbf{C} - (\mathbf{C} : \mathbf{n}) \otimes \frac{\partial \Delta p}{\partial \Delta \varepsilon} - K (\mathbf{I} \otimes \mathbf{I}) \right) \right. \\
& \left. + \left(\frac{-n F_{v_2}}{A_{v_2} \sqrt{\frac{3}{2}} \Delta t} \right) : \left(\frac{2}{3} (A_1 + A_2) \mathbf{n} - \sqrt{\frac{2}{3}} (B_1 \alpha_1 + B_2 \alpha_2) \right) \otimes \frac{\partial \Delta p}{\partial \Delta \varepsilon} \right] \\
& = (n F_v) : \left(\mathbf{C} - (\mathbf{C} : \mathbf{n}) \otimes \frac{\partial \Delta p}{\partial \Delta \varepsilon} - K (\mathbf{I} \otimes \mathbf{I}) \right) \\
& - (n F_v) : \left(\frac{2}{3} (A_1 + A_2) \mathbf{n} - \sqrt{\frac{2}{3}} (B_1 \alpha_1 + B_2 \alpha_2) \right) \otimes \frac{\partial \Delta p}{\partial \Delta \varepsilon} \\
& - \frac{2}{3} F_v (E_1 - F_1 R) \otimes \frac{\partial \Delta p}{\partial \Delta \varepsilon} + (n F_{v_2}) : \left(\mathbf{C} - (\mathbf{C} : \mathbf{n}) \otimes \frac{\partial \Delta p}{\partial \Delta \varepsilon} - K (\mathbf{I} \otimes \mathbf{I}) \right) \\
& - (n F_{v_2}) : \left(\frac{2}{3} (A_1 + A_2) \mathbf{n} - \sqrt{\frac{2}{3}} (B_1 \alpha_1 + B_2 \alpha_2) \right) \otimes \frac{\partial \Delta p}{\partial \Delta \varepsilon} \\
\frac{\partial \Delta p}{\partial \Delta \varepsilon} & = -F_v \left[\mathbf{n} : \left\{ \mathbf{C} : \mathbf{n} + \frac{2}{3} (A_1 + A_2) \mathbf{n} - \sqrt{\frac{2}{3}} (B_1 \alpha_1 + B_2 \alpha_2) \right\} + \frac{2}{3} (E_1 - F_1 R) \right] \otimes \frac{\partial \Delta p}{\partial \Delta \varepsilon} \\
& - F_{v_2} \left[\mathbf{n} : \left\{ \mathbf{C} : \mathbf{n} + \frac{2}{3} (A_1 + A_2) \mathbf{n} - \sqrt{\frac{2}{3}} (B_1 \alpha_1 + B_2 \alpha_2) \right\} \right] \otimes \frac{\partial \Delta p}{\partial \Delta \varepsilon} + F_v \mathbf{n} : [\mathbf{C} - K (\mathbf{I} \otimes \mathbf{I})] \\
& + F_{v_2} \mathbf{n} : [\mathbf{C} - K (\mathbf{I} \otimes \mathbf{I})] \\
& (F_v + F_{v_2}) \mathbf{n} : [\mathbf{C} - K (\mathbf{I} \otimes \mathbf{I})] = \{1 + \mathbb{F}_1 + \mathbb{F}_2\} \otimes \frac{\partial \Delta p}{\partial \Delta \varepsilon}
\end{aligned}$$

Hence,

$$\frac{\partial \Delta p}{\partial \Delta \varepsilon} = \frac{(F_v + F_{v_2}) \mathbf{n} : [\mathbf{C} - K(\mathbf{I} \otimes \mathbf{I})]}{1 + \mathbb{F}_1 + \mathbb{F}_2}$$

Finally, the Jacobian, \mathbf{J} , is

$$\mathbf{J} = \mathbf{C} - (\mathbf{C} : \mathbf{n}) \otimes \frac{(F_v + F_{v_2}) \mathbf{n} : [\mathbf{C} - K(\mathbf{I} \otimes \mathbf{I})]}{1 + \mathbb{F}_1 + \mathbb{F}_2} \quad (5)$$

where

$$\begin{aligned} \mathbb{F}_1 &= F_v \left[\mathbf{n} : \left\{ \mathbf{C} : \mathbf{n} + \frac{2}{3} (A_1 + A_2) \mathbf{n} - \sqrt{\frac{2}{3}} (B_1 \boldsymbol{\alpha}_1 + B_2 \boldsymbol{\alpha}_2) \right\} + \frac{2}{3} (E_1 - F_1 R) \right] \\ \mathbb{F}_2 &= F_{v_2} \left[\mathbf{n} : \left\{ \mathbf{C} : \mathbf{n} + \frac{2}{3} (A_1 + A_2) \mathbf{n} - \sqrt{\frac{2}{3}} (B_1 \boldsymbol{\alpha}_1 + B_2 \boldsymbol{\alpha}_2) \right\} \right] \end{aligned}$$